PSis - Programação de Sistemas - 2016/2017
Exame 1ª Época, 14 de Junho de 2017, 18h30, Duração: 2h

Software Architectures

1. [1.5 points] Describe the main difference between layered and n-tier architectures.

2. [1.5 points] Describe Two advantages of using a P2P architecture when implementaing a certain project.

Operating Systems

3. [1.0 points] Describe the concept of a microkernel. Explain how microkernel architectures increases the reliability of an operating system.

5. [1.0 points] Describe the main difference between a system-call and a regular function.

Processes/ Shared memory /threads

6. [1.0 point] Describe two reasons why a process should perform a **wait** for the death of child processes.

7. [2.0 point] Describe what happens in the shell (terminal) when a user runs a command (for instance **ls**). Use pseudocode to describe the various instructions/functions/system-calls executed by the shell.

8.a) [1.0 point] Unix has a lot of resource shared between different processes (for instance, shared memory, semaphores). Describe two ways to identify (in the C code of the applications and on the operating system) those shared resources.

8.b). [1.0 point] Of those two identifier classes, state which one is more versatile. Describe why.

9.a) [0.5 point] Modern operating system implement memory swap by spawning (writing and reading) process memory to disk. Describe what is the advantage of this memory management approach.

9.b) [1.0 point] What is the mechanism that allows the implementation of swap?

10. **[1.0 point]** In the implementation of a shared mail server (used by multiple users), describe one reason for using multiple processes instead of threads to implement parallelism.

Process Communication

10. [1.0 point] Comparing PIPES with FIFOS, what is the situation where FIFOS should be used? Explain why.

Synchronization

11. [1.5 point] On what situation spin locks should be used, instead of mutexes? Describe why.

12. [2.0 points] Supose we have **4 threads** with the code presented next:

Threads 1..4

```
...
while (1){
    Code_A();
    Code_B();
    Code_C()
}
...
```

Change the code so that:

- **Code_B** from any thread should only start after **Code_A** from all other threads concluded.
- **Code_C** from any thread should only start after **Code_B** from all other threads concluded.
- **Code_A** from any thread should only start after **Code_C** from all other threads concluded.
-
- **Code_A** of all 4 threads can run simultaneous
- No two versions of **Code_B** (from different threads) should run concurrenty
- The last thread finishing **Code_C** should print a message in the screen

Define and initialize the necessary synchronization variables

Software tests

13. [2.0 points] Suppose we plan to develop a **circular buffer** data structure. This data structure can be manipulated by the following functions:

- **`b = create_buffer(size)`** – Creates a buffer with certain maximum size
- **`error = add_data(b, data, length)`** – Adds data of a certain length to the buffer **b**, if data does not fit, return error
- **`len = retrieve_data(b, data, length)`** – retrieves data up to a certain length from the buffer **b**, return the length of retrieved data

Supposing that the unitary tests for the **create_buffer** and **retrieve_data** are working correctly, write **in detain** the various unitary tests for the **add_data** function.
Write the various pseudo-code tests and describe for each piece of code what it tests in the **add_data** function.

14. [1.0 point] Decomposability is a feature that the various software architectures try to achieve. Describe what is Decomposability.
Describe on what way a good decomposability of a system eases the testing procedures?